Remarks

Applicants respectfully request reconsideration of the present application in view of the foregoing amendments and the following remarks. Claims 7-11 and 18-24 are pending in the application. Claims 7-11 and 18-24 are rejected. No claims have been allowed. Claims 7, 18 and 21 are independent.

Substance of Interview

Applicant's Representative thanks the Examiner for the telephonic interview of May 18, 2009. Claim 7 was discussed with proposed amendments. No agreement was reached. However, Applicant's Representative included language into the independent claims regarding disconnecting a physical hardware connection. The prior art appears to describe closing a socket, but does not completely disconnect the hardware connection.

Cited Art

The Action cites Microsoft, Windows Sockets 2 Service Provider Interface, Revision 2.2.2, August 7, 1997 (Winsock 2.2.2); Gase, U.S. Patent No. 6,363,081 (Gase); Stone, U.S. Patent No. 5,802,304 (Stone); Morris et al., U.S. Patent No. 7,069,333 (Morris); and Hong et al., U.S. Patent No. H2065H (Hong).

Claim Rejections under 35 U.S.C. § 112

The Action rejects claim 7 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

In response, Applicant has amended claim 7 to place the claim in the desired format for the Examiner. Applicant's representative does believe that this amendment was unnecessary, but wishes to expedite prosecution, and therefore has agreed to make the amendment.

Claim Rejections under 35 U.S.C. § 103

The Action rejects claims 7, 8, 18, 19, 21, 22, and 23 under 35 U.S.C. 103(a) unpatentable over Winsock 2.2.2, in further view of Gase.

Applicant's representative notes that the Winsock 2.2.2. document does not appear to have been published or at least the publication date is unclear. For example, the first few pages of the table of contents do not even properly list page numbers, indicating that it was not yet ready to be published. Merely having a revision date does not mean that the Winsock 2.2.2. document was publicly available on that date. Most likely, the revision date is the last day that anyone changed the document and it could have been published years later. Applicant's representative kindly requests that proper publication information be provided to support that the Winsock 2.2.2. document was publicly available before the priority date of the present application.

Additionally, it is noted that Winsock 2.2.2 teaches away from using a connection manager. In particular, Winsock 2.2.2. states the following:

> However, the Winsock interface does not implement any type of access control, so it is up to the processes involved to coordinate their operations on a shared socket. <u>A typical use for shared sockets is to have one process that is responsible for creating sockets and establishing connections, hand off sockets to other processes which are responsible for information exchange</u>.

Paragraph 2.10 (page 17)(emphasis added)

Winsock 2.2.2. section 4.29 has similar language under the "comments" section: "However, the WinSock interface does not implement any type of access control, so it is up to the processes involved to coordinate their operations on a shared socket".

Claim 7, by contrast, requires calling a connection manager to establish the connection from both the first application and the second application:

"receiving, from a first application on a client computer, a first request, ~~in~~ <u>at</u> a connection manager, for a connection to a remote resource;" and

"receiving, ~~in~~ <u>at</u> the connection manager, a second request from a second application for connection to the same remote resource as the first application."

Thus, each application in claim 7 that wants to establish a connection uses the connection manager. Winsock 2.2.2. clearly does not use a connection manager, as one of the shared processes must hand off sockets to another of the shared processes (see quote above).

However, there is another difference captured in claim 7. Specifically, disconnecting results in a disconnection of the physical hardware connection. Support for this amendment can

be found on page 10 of the original application, lines 10-23. One example is where there is a phone connection, the physical connection will be disconnected.

There is still another difference captured in claim 7. Applicant's representative believes that Winsock 2.2.2. is directed to an API that assumes a hardware connection is already established to the network. Indeed, the title is called Windows Sockets 2 Application Programming Interface. Also, Section 1.3, second paragraph, states that "[t]his document comprises only the API portion of the Windows Sockets 2 specification." Thus, Winsock 2.2.2. has nothing to do with establishing a hardware connection in the first place and then sharing the physical hardware connection. Rather, Winsock 2.2.2. describes how to create a software socket.

Comparing this to Figure 3 of the present application, the API is similar to Wininet 304. However, claim 7 is not directed to sharing at the API level, but rather at hardware level. One example of establishing a hardware connection is using the dialer 316 of Figure 3. Claim 7 now requires "wherein sharing the connection includes having the first and second applications using the same physical hardware connection to the remote resource." Claim 7 also requires creating the connection between the first application and the remote resource when physical hardware connection between the client computer and the remote is not already established (emphasis added). Winsock 2.2.2. does not relate to creating a physical hardware connection when one is not already established. Instead, Applicant's representative believes that Winsock 2.2.2. assumes that a hardware connection is already established. The Examiner argues that Winsock 2.2.2. discusses, in section 4.53, that WSASocket creates a socket and therefore a connection. While WSASocket allows a socket to be created (which is a software window to allow communication over a network), the physical hardware connection is assumed to exist already.

Gase discloses a system where a primary application is connected first and then a monitoring application submits a registration request to the primary application over a registration port. (Col. 4, lines 51-56). When "the primary applications receive data packets over their contested ports, they retransmit the packets to the monitoring application 56." (Col. 4, lines 57-60). The registration ports are shown at 58(1) – 58(3) in Figure 2 of Gase.

First, Gase does not use a connection manager. Instead, in Gase, a monitoring application has to request that a primary application retransmit packets to the monitoring application. Second, Gase clearly requires retransmission from one application to another: the primary application retransmits the packets to the monitoring application 56. (Col. 4,

lines 57-60). Thus, the monitoring application does not even appear to be connected to a remote resource.

Therefore, Gase has the same deficiencies as the Winsock application. Each teach away from using a connection manager. Claim 7 is, therefore, in condition for allowance and such action is respectfully requested.

Claim 10 depends from claim 7 and should be in condition for allowance for the reasons stated above. Additionally, claim 10 requires the specific hardware connection to be a dial-up connection in order to further emphasize that a physical hardware connection occurs as a result of a request.

Claim 18 also uses a "connection manager" in order to manage the processes as they connect and disconnect from the remote resource. As described above, Winsock 2.2.2. does not use a connection manager, but instead, as stated in section 4.29, under the "comments" section: "the WinSock interface does not implement any type of access control, so it is up to the processes involved to coordinate their operations on a shared socket" (emphasis added). Thus, claim 18 requires a connection manager, which is used to provide a centralized access control, which is exactly what Winsock 2.2.2. teaches against. Additionally, claim 18 requires creating a physical hardware connection in response to a request to emphasize that claim 18 is at a hardware layer, not at a high-level API as described in the Winsock 2.2.2. reference. Finally, claim 18 requires "disconnecting the physical hardware connection when a process requests a disconnection when stored identifiers indicate no other process is communicating with remote resources via the connection". Thus, the physical hardware connection is terminated.

Claim 21 also requires establishing a physical hardware connection in response to the request from a first application. Additionally, claim 21 requires the centralized connection manager, which is what Winsock 2.2.2. teaches against, as described above.

The remaining claims depend from the independent claims and should be in condition for allowance for the reasons stated above. The remaining cited references were merely used to show dependent claims and the rejection based on these references is moot in view of the above-mentioned amendments.

Interview Request

If the claims are not found by the Examiner to be allowable, the Examiner is requested to call the undersigned attorney to set up an interview to discuss this application.
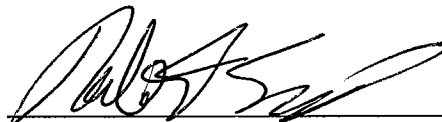
Conclusion

The claims in their present form should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By _____
Robert F. Scotti
Registration No. 39,830